

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP2005/018015

International filing date: 29 September 2005 (29.09.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-286412
Filing date: 30 September 2004 (30.09.2004)

Date of receipt at the International Bureau: 17 November 2005 (17.11.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2 0 0 4 年 9 月 3 0 日

出 願 番 号
Application Number: 特 願 2 0 0 4 - 2 8 6 4 1 2

パリ条約による外国への出願
に用いる優先権の主張の基礎
となる出願の国コードと出願
番号
J P 2 0 0 4 - 2 8 6 4 1 2
The country code and number
of your priority application,
to be used for filing abroad
under the Paris Convention, is

出 願 人
Applicant(s): 日 本 電 信 電 話 株 式 有 限 公 司

2 0 0 5 年 1 1 月 2 日

特許庁長官
Commissioner,
Japan Patent Office

中 嶋



【書類名】	特許願
【整理番号】	NTTH165957
【提出日】	平成16年 9月30日
【あて先】	特許庁長官殿
【国際特許分類】	H04N 7/32
【発明者】	
【住所又は居所】	東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内
【氏名】	高村 誠之
【発明者】	
【住所又は居所】	東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内
【氏名】	八島 由幸
【特許出願人】	
【識別番号】	000004226
【氏名又は名称】	日本電信電話株式会社
【代理人】	
【識別番号】	100087848
【弁理士】	
【氏名又は名称】	小笠原 吉義
【電話番号】	03-3807-1151
【選任した代理人】	
【識別番号】	100074848
【弁理士】	
【氏名又は名称】	森田 寛
【選任した代理人】	
【識別番号】	100095072
【弁理士】	
【氏名又は名称】	岡田 光由
【手数料の表示】	
【予納台帳番号】	012586
【納付金額】	16,000円
【提出物件の目録】	
【物件名】	特許請求の範囲 1
【物件名】	明細書 1
【物件名】	図面 1
【物件名】	要約書 1
【包括委任状番号】	0005321

【書類名】 特許請求の範囲

【請求項 1】

原信号に一致する復号が可能なビデオ符号化を行う可逆ビデオ符号化方法であって、
あらかじめ定められた非可逆のビデオ符号化方式に準拠し、画像信号の各ブロック毎に
原信号からフレーム内符号化における空間予測またはフレーム間符号化における時間予測
による予測信号を差し引いた残差信号を入力するステップと、

前記非可逆のビデオ符号化方式に基づき、前記残差信号に対し直交変換を施して得られ
る変換係数およびそれを量子化した量子化係数を求めるステップと、

前記量子化係数、量子化時に用いた量子化パラメータおよび量子化方法から定まる変換
係数の存在空間を特定するステップと、

前記変換係数の存在空間内のある格子点が残差信号の直交変換の結果として妥当である
かどうかを判断する妥当性判断のステップと、

前記変換係数の存在空間内の格子点のうち前記判断が妥当であるものを所定の格子点の
順番で探索し列挙するステップと、

列挙された格子点に、列挙順に通し番号を割り当てるステップと、

列挙された格子点の中で、残差信号の変換係数に一致する格子点の通し番号を得るステ
ップと、

前記残差信号の変換係数に一致する格子点の通し番号を符号化し出力するステップとを
有する

ことを特徴とする段階的可逆ビデオ符号化方法。

【請求項 2】

請求項 1 に記載の段階的可逆ビデオ符号化方法において、

前記変換係数の存在空間内の格子点のうち前記判断が妥当であるものを列挙するステッ
プでは、変換係数間の整数的値関係を用いることにより、前記空間内で変換係数がとり得
ない点については、残差信号の直交変換の結果として妥当であるかどうかの判断処理を省
略する

ことを特徴とする段階的可逆ビデオ符号化方法。

【請求項 3】

請求項 1 または請求項 2 に記載の段階的可逆ビデオ符号化方法において、

特定のいくつかの変換係数について、該係数と既に出力済みの係数との整数的値関係
を利用し、該係数がとり得ない値を除いた情報を出力することで、該係数を出力することに
代替するステップを有し、

前記変換係数の存在空間内の格子点のうち前記判断が妥当であるものを列挙するステッ
プでは、前記あらかじめ出力された変換係数を用い、次元の縮小した存在空間内で格子点
を列挙する

ことを特徴とする段階的可逆ビデオ符号化方法。

【請求項 4】

請求項 1 から請求項 3 までのいずれか 1 項に記載の段階的可逆ビデオ符号化方法におい
て、

前記妥当性判断のステップでは、ビット演算および整数加減算のみを用いて妥当性を判
断する

ことを特徴とする段階的可逆ビデオ符号化方法。

【請求項 5】

請求項 1 から請求項 4 までのいずれか 1 項に記載の段階的可逆ビデオ符号化方法におい
て、

残差信号の存在範囲と変換係数の存在範囲の交わりが凸多面体となることを用いて、判
断対象となる前記格子点が前記凸多面体の外部に存在する場合に前記妥当性判断を途中で
中止する

ことを特徴とする段階的可逆ビデオ符号化方法。

【請求項 6】

請求項 1 から請求項 3 までのいずれか 1 項に記載の段階的可逆ビデオ符号化方法において、

前記変換係数の存在空間内の格子点のうち前記判断が妥当であるものを列挙するステップを実行せずに符号量の見積もりを行うステップを有する

ことを特徴とする段階的可逆ビデオ符号化方法。

【請求項 7】

請求項 1 から請求項 5 までのいずれか 1 項に記載の段階的可逆ビデオ符号化方法により符号化された符号化ストリームを復号する復号方法であって、

前記あらかじめ定められた非可逆のビデオ符号化方式に対応する復号方式を実行するステップと、

量子化パラメータ、量子化係数および符号化側の量子化方法、ならびに既に復号された係数がある場合にはその係数から定まる変換係数の存在空間を特定するステップと、

復号すべき係数を復号するステップと、

前記変換係数の存在空間内のある格子点が残差信号の直交変換の結果として妥当であるかどうかを判断する妥当性判断のステップと、

前記変換係数の存在空間内の格子点のうち前記判断が妥当であるものを、符号化時における格子点の探索の順番と同じ順番で探索し列挙するステップと、

通し番号を復号するステップと、

前記列挙された格子点の中で、前記復号された通し番号に等しい順番の格子点を出力するステップとを有する

ことを特徴とする段階的可逆ビデオ復号方法。

【請求項 8】

原信号に一致する復号が可能なビデオ符号化を行う可逆ビデオ符号化装置であって、

あらかじめ定められた非可逆のビデオ符号化方式に準拠し、画像信号の各ブロック毎に原信号からフレーム内符号化における空間予測またはフレーム間符号化における時間予測による予測信号を差し引いた残差信号を入力する手段と、

前記非可逆のビデオ符号化方式に基づき、前記残差信号に対し直交変換を施して得られる変換係数およびそれを量子化した量子化係数を求める手段と、

前記量子化係数、量子化時に用いた量子化パラメータおよび量子化方法から定まる変換係数の存在空間を特定する存在空間決定手段と、

前記変換係数の存在空間内のある格子点が残差信号の直交変換の結果として妥当であるかどうかを判断する妥当性判断手段と、

前記変換係数の存在空間内の格子点のうち前記判断が妥当であるものを所定の格子点の順番で探索し列挙する手段と、

列挙された格子点に、列挙順に通し番号を割り当てる手段と、

列挙された格子点の中で、残差信号の変換係数に一致する格子点の通し番号を得る手段と、

前記残差信号の変換係数に一致する格子点の通し番号を符号化し出力する手段とを備える

ことを特徴とする段階的可逆ビデオ符号化装置。

【請求項 9】

請求項 8 に記載の段階的可逆ビデオ符号化装置により符号化された符号化ストリームを復号する復号装置であって、

前記あらかじめ定められた非可逆のビデオ符号化方式に対応する復号方式を実行する手段と、

量子化パラメータ、量子化係数および符号化側の量子化方法、ならびに既に復号された係数がある場合にはその係数から定まる変換係数の存在空間を特定する存在空間決定手段と、

復号すべき係数を復号する係数復号手段と、

前記変換係数の存在空間内のある格子点が残差信号の直交変換の結果として妥当である

かどうかを判断する妥当性判断手段と、

前記変換係数の存在空間内の格子点のうち前記判断が妥当であるものを、符号化時における格子点の探索の順番と同じ順番で探索し列挙する手段と、

通し番号を復号する通し番号復号手段と、

前記列挙された格子点の中で、前記復号された通し番号に等しい順番の格子点を出力する出力手段とを備える

ことを特徴とする段階的可逆ビデオ復号装置。

【請求項 1 0】

請求項 1 から請求項 6 までのいずれか 1 項に記載の段階的可逆ビデオ符号化方法をコンピュータに実行させるための段階的可逆ビデオ符号化プログラム。

【請求項 1 1】

請求項 7 に記載の段階的可逆ビデオ復号方法をコンピュータに実行させるための段階的可逆ビデオ復号プログラム。

【請求項 1 2】

請求項 1 から請求項 6 までのいずれか 1 項に記載の段階的可逆ビデオ符号化方法をコンピュータに実行させるためのプログラムをコンピュータ読み取り可能な記録媒体に記録したことを特徴とする段階的可逆ビデオ符号化プログラムの記録媒体。

【請求項 1 3】

請求項 7 に記載の段階的可逆ビデオ復号方法をコンピュータに実行させるためのプログラムをコンピュータ読み取り可能な記録媒体に記録したことを特徴とする段階的可逆ビデオ復号プログラムの記録媒体。

【書類名】 明細書

【発明の名称】 段階的可逆ビデオ符号化方法，段階的可逆ビデオ復号方法，段階的可逆ビデオ符号化装置，段階的可逆ビデオ復号装置，これらのプログラムおよびそのプログラムの記録媒体

【技術分野】

【0001】

本発明は高能率ビデオ符号化／復号技術に関し，特に，基本部分として伝送される符号はH．264標準と互換性を保ちながら，付加部分の符号量をできるだけ低く抑えつつ，原信号に一致する可逆な復号を可能とする段階的可逆ビデオ符号化／復号技術に関する。

【背景技術】

【0002】

従来の可逆あるいは段階的ビデオ符号化方式には以下のようなものがある。Motion JPEG 2000標準（非特許文献1参照）の可逆モードやJPEG-L S（非特許文献2参照）は，静止画ベースであるためフレーム内に閉じた符号化を行う。

【0003】

H．264標準（非特許文献3参照）の第二版“Fidelity Range Extension(FRExt)”は，フレーム内・フレーム間予測残差信号を（直交変換・量子化せず）そのまま伝送する。

【0004】

非可逆な符号化方式と組み合わせた段階的符号化方式としては，復号画像と原画像の差分を符号化するもの（非特許文献4参照）がある。これは基本情報にMP E G－2を用いているが，他の方式であっても適用は可能である。

【0005】

また，MP E G－4標準（非特許文献5参照）のFine Granularity Scalable(FGS) Profile方式のように，D C T（離散コサイン変換）を施し整数化したものに対し，ビットプレーン展開し逐次伝送するというものがある。

【非特許文献1】 ISO/IEC 15444-3:2002 Information technology — JPEG 2000 image coding system — Part 3: Motion JPEG 2000

【非特許文献2】 ISO/IEC 14495-1:1999 Information technology — Lossless and near-lossless coding of continuous tone still images

【非特許文献3】 ISO/IEC 14496-10:2003 Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding

【非特許文献4】 中嶋，八島，小林：“MP E G－2符号化パラメータに基づく階層的ロスレス符号化の検討”，信学総大 D-11-49，Mar. 2000

【非特許文献5】 ISO/IEC 14496-2:2003 Information technology — Coding of audio-visual objects — Part 2: Visual

【発明の開示】

【発明が解決しようとする課題】

【0006】

上記J P E G 2 0 0 0（非特許文献1参照）は，段階的伝送が可能であるが，フレーム内符号化をするため，ビデオ特有のフレーム間相関を用いた高能率符号化はできないという問題がある。また，上記J P E G－L S（非特許文献2参照）は，J P E G 2 0 0 0よりも効率は高いが，やはりフレーム内に閉じた符号化であり効率に限界があるほか，段階的伝送ができないという問題がある。

【0007】

H．264標準（非特許文献3参照）の第二版“Fidelity Range Extension(FRExt)”は，段階的伝送ができないという問題がある。

【0008】

また，上記非特許文献4に記載された技術は，直交変換空間内での残差ではなく，原信号空間での残差を符号化対象とするため，元来原信号が存在しないはずの空間までも考慮

した符号化をせざるをえず、圧縮効率に限界がある。

【0009】

MPEG-4 標準（非特許文献5 参照）の Fine Granularity Scalable (FGS) Profile 方式は、

- ・実数変換である DCT 後、係数が整数化されるため、いくら付加情報を多く使っても可逆にはできないという問題、
 - ・後述の式（6）のように変換後の係数が伸張される H. 264 標準方式においてそのまま適用すると、伸張分がそのまま符号量の無駄につながってしまうという問題、
- などがあった。

【0010】

このようにフレーム間予測を行うことでビデオ符号化効率を高め、スケーラビリティを有し、かつ基本情報が H. 264 標準互換である方式は提案されていなかった。

【0011】

本発明は、上に述べたような問題に鑑みて、基本部分として伝送される符号は H. 264 標準と互換性を保ちながら、付加部分の符号量をできるだけ低く抑えつつ、可逆な復号を可能とすることを目的とする。

【課題を解決するための手段】

【0012】

上記課題を解決するため、第1の本発明に係る段階的可逆ビデオ符号化方式は、H. 264 標準準拠のビットストリームを生成する手段と、これに加えて画像信号の各ブロックの符号化において、フレーム内符号化における空間予測またはフレーム間符号化における時間予測による「予測信号」を入力する手段と、原信号から予測信号を差し引いた「残差信号」を入力する手段と、H. 264 標準方式に基づき残差信号に対し直交変換を施して得られる「変換係数」およびそれを量子化した「量子化係数」を求める手段と、量子化係数、量子化パラメータおよび量子化方法から定まる「変換係数の存在空間」を特定する手段と、該空間中のある格子点が残差信号の直交変換の結果として妥当であるかどうかを判断する「妥当性判断手段」と、該空間内の格子点のうち上記判断が妥当であるものを列挙する「列挙手段」と、列挙された格子点に、列挙順に0から始まる「通し番号」を割り当てる手段と、列挙された格子点の中で、残差信号の変換係数に一致する格子点の通し番号を得る手段と、得られた通し番号を符号化する手段とを備えることを特徴とする。

【0013】

第2の本発明は、第1の本発明において、格子点を列挙する際に、変換係数間の整数的値関係を用いることにより、空間内で変換係数がとりえない点については妥当の判断処理を省略する手段を備えることを特徴とする。

【0014】

第3の本発明は、第1または第2の本発明において、特定のいくつかの変換係数について、該係数と既に伝送済みの係数（もしあれば）との整数的値関係を利用し、該係数がとりえない値を除いた情報を伝送することで、該係数を伝送することに代替する手段と、こうして予め伝送された変換係数を用い、次元の縮小した存在空間内で格子点を列挙する手段とを備えることを特徴とする。

【0015】

第4の本発明は、第1～第3の本発明において、前記妥当性判断をビット演算・整数加減算のみで等価に実現する手段を備えることを特徴とする。

【0016】

第5の本発明は、第1～第4の本発明において、残差信号の存在範囲と変換係数の存在範囲の交わりが凸多面体となることを用いて上記「妥当性判断」を途中で中止する手段を備えることを特徴とする。

【0017】

第6の本発明は、第1～第3の本発明において、上記列挙手段による格子点の列挙を実行せず符号量の見積もりを行う手段を備えることを特徴とする。

【0018】

第7の本発明は、第1～第5の本発明による段階的可逆ビデオ符号化方式に対応する復号方式であって、H. 264標準復号方式を実行する手段と、量子化パラメータ、量子化係数、符号化側の量子化方法、および既に復元された係数（もしあれば）から定まる変換係数の存在空間を特定する手段と、復号すべき係数を復号する手段と、第1～第3の本発明と同様な列挙手段および妥当性判断手段と、通し番号を復号する手段と、列挙された順が通し番号に等しい格子点を出力する手段とを備えることを特徴とする。

【0019】

以上の段階的可逆ビデオ符号化および復号の処理は、コンピュータとソフトウェアプログラムとによって実現することもでき、そのプログラムをコンピュータ読み取り可能な記録媒体に記録して提供することも、ネットワークを通して提供することも可能である。

【発明の効果】

【0020】

本発明によれば、基本部分として伝送される符号はH. 264標準の符号化と互換性を保ちながら、付加部分の符号量をできるだけ低く抑えつつ、原信号に一致する可逆な復号を可能とすることができる。

【0021】

また、本発明によれば、処理を数兆倍に高速化して上記の処理を実行することができる。また、実際の符号化を行わずに、符号量の推定を行うことができ、結果として符号量を減らす予測モードの選択を高速に行うことができる。

【発明を実施するための最良の形態】

【0022】

本発明の説明の前提として、H. 264方式の直交変換について説明する。

〔H. 264方式の直交変換〕

H. 264標準においては、フレーム内あるいはフレーム間で画素値を予測した後、縦横4画素ずつの小ブロック毎に、残差の直交変換・係数の量子化を行う。

【0023】

原信号の小ブロックを4×4行列Uで、フレーム内あるいはフレーム間で該ブロックを予測した信号を同じく4×4行列Yで表す。そして予測残差信号（4×4行列R）を

$$R = U - Y \quad (1)$$

とする。これらはいずれも、要素がすべて整数である。ここでは残差信号の各要素を

【0024】

【数1】

$$R = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \quad (2)$$

のように表記することとする。

【0025】

これに次のような直交変換を施す。

【0026】

$$X = T R T^t \quad (3)$$

ここで

【0027】

【数 2】

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad (4)$$

である。また、 T^t は行列 T の転置行列を表す。

【0 0 2 8】

さらに、 X の各要素を

【0 0 2 9】

【数 3】

$$X = \begin{bmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ M & N & O & P \end{bmatrix} \quad (5)$$

とする。

【0 0 3 0】

ここで T は直交変換ではあるが正規直交変換ではないことに注意する。正規直交変換の行列式は常に 1 であるが、 T の行列式 ($\det T$) は 4 0 であるので、任意の 1 6 次元領域の体積は T による写像後 4 0 倍になる。式 (3) は T により R の 4 行および 4 列に変換を施すため、変換後の係数 X は、変換前 (残差信号 R) に比べ

$$4 \cdot 0^{4+4} = 6,553,600,000,000 \text{ 倍} \quad (6)$$

に引き伸ばされた、極めて疎な空間内に写像される。H. 2 6 4 における直交変換係数は整数となるが、係数空間の格子点は、ほとんどが残差信号として不適 (つまり後の式 (8) により逆変換しても整数値が得られない) であるということになる。仮に係数空間の格子点をすべて対象とし、原信号に対応する格子点を符号化すると、上記の数の 2 を底とする対数を画素数 ($4 * 4 = 16$) で除した

$$\log_2 (4 \cdot 0^{4+4}) / 16 = 2.66 \text{ [bit]} \quad (7)$$

が、1 画素あたり余分に必要になる。対象としている信号が (1 カラーコンポーネントあたり) 8 bit であることを考えると、2.66 bit の増加は実用性を著しく阻害していると言わざるを得ない。

【0 0 3 1】

H. 2 6 4 方式では、前述のように大きく拡大された X の各要素を、粗く (比較的広い幅で) 量子化することにより、この拡大分を補償している。

【0 0 3 2】

もし量子化がなければ、逆変換

$$R = T^{-1} X (T^t)^{-1} \quad (8)$$

により残差信号は完全に元に戻り、

$$U = R + Y \quad (9)$$

により原画素値の 4×4 行列 U が再現できるところであるが、実際の H. 2 6 4 符号化においては量子化を施された係数が復号側で逆量子化されると、一般に、元の値には近いが完全に同じではない値が復元されることになる。こうして X は伝送後、異なる値 X' ($\neq X$) となり、これを式 (8) により逆変換しても

$$R \neq R' = T^{-1} X' (T^t)^{-1}$$

となる。復号側では Y は符号化側と同一のものを持つことができるが、 R が再現できない

ため原画素値 $U = R + Y$ も再現できない。

【0033】

したがって U を完全再現するためには、 X の量子化で失われた情報を補うような付加情報を、別途伝送する必要がある。

【0034】

〔第1の発明〕

第1の発明に係る段階的可逆ビデオ符号化方法は、この量子化で失われた情報を、以下のように効率よく符号化する。

【0035】

例えば、式(5)における係数 A の値は量子化され伝送されるため、復号側では量子化前の A の値は正確にはわからない。ここでは符号化側の量子化方法は既知としており、 A がとり得る範囲はわかる。すなわち、H. 264 ソフトウェア JM (参考文献: <http://bs.hhi.de/~suehring/tml/>, “JM Reference Software version 8.4”, Jul 2004) の量子化方法を例にとると、

$$| \text{level}_A | = (| A | * q_{c_A} + qpconst) >> qbits \quad (10)$$

のようにして量子化された値 level_A を求める。ここで “ $>>$ ” は右ビットシフト演算子である。なお level_A の正負は A のそれに一致させる。ここで q_{c_A} は A の位置、量子化パラメータに対応して JM が決める整数、 $qpconst$ は符号化モード、量子化パラメータに対応して JM が決める整数、 $qbits$ は量子化パラメータに対応して、JM が決める整数である。

【0036】

復号側では量子化パラメータや符号化モードおよび量子化方法を知ることができるので、符号化側と共通の level_A 、 $qpconst$ 、 $qbits$ を持つことができる。 level_A の値からまず

$$x = \text{level}_A << qbits$$

$$y = x + (1 << qbits) - 1$$

$$mmin = (x - qpconst + q_{c_A} - 1) / q_{c_A}$$

$$mmax = (y - qpconst) / q_{c_A}$$

を求める。ここで “ $<<$ ” は左ビットシフト演算子、“ $/$ ” は小数点以下を切捨てる整数除算である。ついで上下限 A_{\min} 、 A_{\max} が以下のように求められる。

【0037】

【数4】

$$A_{\min} = \begin{cases} -mmax & (\text{level}_A < 0 \text{ のとき}) \\ -mmax & (\text{level}_A = 0 \text{ のとき}) \\ mmin & (\text{level}_A > 0 \text{ のとき}) \end{cases} \quad (11)$$

$$A_{\max} = \begin{cases} -mmin & (\text{level}_A < 0 \text{ のとき}) \\ mmax & (\text{level}_A = 0 \text{ のとき}) \\ mmax & (\text{level}_A > 0 \text{ のとき}) \end{cases} \quad (12)$$

こうして、量子化前の係数がとり得る上下限值

$$\text{下限値 } A_{\min}, B_{\min}, C_{\min}, \dots, P_{\min} \quad (13)$$

$$\text{上限値 } A_{\max}, B_{\max}, C_{\max}, \dots, P_{\max} \quad (14)$$

を得ることができる。これらの数値範囲が第1の発明の「変換係数の存在空間」に相当する。

【0038】

この上下限を元に、下記のアлゴリズム1のような16重ループ(第1の発明の「列挙

手段」に相当)を回すことにより, 直交変換後の空間において妥当なものをもれなくすべて列挙することができる。ここで, 以下の処理が第1の発明の「妥当性判断」に相当する。

- ・「Rの全要素が整数である」
- ・「Uの全要素が8bit整数の範囲[0~255]におさまっている」

上記の妥当性判断の結果, 列挙された格子点の総数はcases, 残差信号の直交変換係数に一致する格子点の「通し番号」はindexとして与えられる。

【0039】

(アルゴリズム1)

```
1.  index ← 0
2.  cases ← 0
3.  for A ← Amin to Amax
4.  for B ← Bmin to Bmax
5.  for C ← Cmin to Cmax
6.  for D ← Dmin to Dmax
7.  for E ← Emin to Emax
8.  for F ← Fmin to Fmax
9.  for G ← Gmin to Gmax
10. for H ← Hmin to Hmax
11. for I ← Imin to Imax
12. for J ← Jmin to Jmax
13. for K ← Kmin to Kmax
14. for L ← Lmin to Lmax
15. for M ← Mmin to Mmax
16. for N ← Nmin to Nmax
17. for O ← Omin to Omax
18. for P ← Pmin to Pmax
19.  begin
20.    R ← T-1 X (Tt)-1 (*式(8)*)
21.    if Rの全要素が整数である
22.      then (*A~Pが整数信号の直交変換の結果として妥当*)
23.        U ← R + Y (*式(9)*)
24.        if Uの全要素が8bit整数の範囲[0~255]におさまっている
25.          then (*A~Pが残差信号の直交変換の結果として妥当*)
26.            if Xが原変換係数に一致する
27.              then index ← cases
28.            endif
29.          cases ← cases + 1
30.        endif
31.      endif
32.    end
33.  cases分の情報量を用いてindexを符号化
```

【第2の発明】: 変換係数間の整数的値関係を利用した高速化

第1の発明では, 16重のループをそれぞれ1間隔で回すため, 総ループ回数は非常に多くなる。しかし, 第2の発明では, 直交変換係数間に存在する整数的関係を用いることで, 符号化効率は同じままで, ループ回数を大幅に削減することができる。

【0040】

まず4×4行列Rを, 上の行から下の行へ順に並べなおした16次元の行ベクトル↑xにより

↑x = [a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p]

のように表現することとする。

【 0 0 4 1 】

式 (3) をこの 1 6 次元ベクトルを用いて書き換えると、

$$A = \uparrow t_A \uparrow x^t$$

$$B = \uparrow t_B \uparrow x^t$$

...

$$P = \uparrow t_P \uparrow x^t$$

のように書くことができる。

ここで

$$\uparrow t_A = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$$

$$\uparrow t_B = [2, 1, -1, -2, 2, 1, -1, -2, 2, 1, -1, -2, 2, 1, -1, -2]$$

$$\uparrow t_C = [1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1]$$

$$\uparrow t_D = [1, -2, 2, -1, 1, -2, 2, -1, 1, -2, 2, -1, 1, -2, 2, -1]$$

$$\uparrow t_E = [2, 2, 2, 2, 1, 1, 1, 1, -1, -1, -1, -1, -2, -2, -2, -2]$$

$$\uparrow t_F = [4, 2, -2, -4, 2, 1, -1, -2, -2, -1, 1, 2, -4, -2, 2, 4]$$

$$\uparrow t_G = [2, -2, -2, 2, 1, -1, -1, 1, -1, 1, 1, -1, -2, 2, 2, -2]$$

$$\uparrow t_H = [2, -4, 4, -2, 1, -2, 2, -1, -1, 2, -2, 1, -2, 4, -4, 2]$$

$$\uparrow t_I = [1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1]$$

$$\uparrow t_J = [2, 1, -1, -2, -2, -1, 1, 2, -2, -1, 1, 2, 2, 1, -1, -2]$$

$$\uparrow t_K = [1, -1, -1, 1, -1, 1, 1, -1, -1, 1, 1, -1, 1, -1, -1, 1]$$

$$\uparrow t_L = [1, -2, 2, -1, -1, 2, -2, 1, -1, 2, -2, 1, 1, -2, 2, -1]$$

$$\uparrow t_M = [1, 1, 1, 1, -2, -2, -2, -2, 2, 2, 2, 2, -1, -1, -1, -1]$$

$$\uparrow t_N = [2, 1, -1, -2, -4, -2, 2, 4, 4, 2, -2, -4, -2, -1, 1, 2]$$

$$\uparrow t_O = [1, -1, -1, 1, -2, 2, 2, -2, 2, -2, -2, 2, -1, 1, 1, -1]$$

$$\uparrow t_P = [1, -2, 2, -1, -2, 4, -4, 2, 2, -4, 4, -2, -1, 2, -2, 1]$$

である。

【 0 0 4 2 】

ここで $\uparrow t_A + \uparrow t_C$ を計算すると

$$\begin{aligned} \uparrow t_A + \uparrow t_C &= [2, 0, 0, 2, 2, 0, 0, 2, 2, 0, 0, 2, 2, 0, 0, 2] \\ &= 2 [1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1] \end{aligned}$$

となる。したがって、

$$\begin{aligned} A + C &= (\uparrow t_A + \uparrow t_C) \uparrow x^t \\ &= 2 [1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1] \uparrow x^t \end{aligned}$$

であるので、任意の整数要素ベクトル $\uparrow x$ に対し、 $A + C$ は常に偶数となる。

【 0 0 4 3 】

つまり A の値がわかっている場合、 C の下位第 1 ビットは A のそれと同じとわかるので、 C の存在範囲は、

$$C_{\min} + ((C_{\min} + A) \& 1) \leq C \leq C_{\max} - ((C_{\max} + A) \& 1) \quad (16)$$

であり、この区間中に間隔 2 でまばらに存在していることになる。ここで ' $\&$ ' はビット AND 演算子である。同じ位置関係である、

E と G

I と K

M と O

も同様である。

【 0 0 4 4 】

同様に縦方向においても、 $A + I$ も偶数であることから、 A の値がわかっている場合、 I が間隔 2 で存在する範囲がわかる。同じ位置関係である、

B と J

C と K

D と L

も同様である。

【0045】

次に、任意の整数要素ベクトル $\uparrow x$ に対し、 $B + (C \gg 1) + (A \gg 1)$ は常に偶数になる。これは、 $\uparrow x$ の各要素を 0, 1, 2, 3 と変化させたすべての場合について確認することで証明される。つまり A と C の値がわかっている場合、B の下位第 1 ビットは $(C \gg 1) + (A \gg 1)$ のそれと同じとわかるので、上記と同様に B の存在範囲がわかり、そこに間隔 2 で存在することになる。同じ位置関係、および縦に同じ位置関係である、

F と $(E \gg 1) + (G \gg 1)$

J と $(I \gg 1) + (K \gg 1)$

N と $(M \gg 1) + (O \gg 1)$

E と $(A \gg 1) + (I \gg 1)$

F と $(B \gg 1) + (J \gg 1)$

G と $(C \gg 1) + (K \gg 1)$

H と $(D \gg 1) + (L \gg 1)$

も同様である。

【0046】

次に

$$\begin{aligned} 2 \cdot 5 (\uparrow t_A + \uparrow t_C) + 2 \uparrow t_B + \uparrow t_D \\ = [10, 0, 0, 0, 10, 0, 0, 0, 10, 0, 0, 0, 10, 0, 0, 0] \\ = 10 [1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0] \end{aligned}$$

となるので、任意の整数要素ベクトル $\uparrow x$ に対し、 $2B + D + 2 \cdot 5(A + C)$ は、常に 10 の倍数になる。

【0047】

つまり A, B, C の値がわかっている場合、D の 10 を法とする剰余 ($D \bmod 10$) と $2 \cdot 5(A + C) + 2B$ のそれとの和は 0 か 10 となるので、D の存在範囲がわかり、その中に間隔 10 で存在することになる。

【0048】

同じ位置関係や、縦に同じ位置関係、例えば M と A, E, I の間にも同様の関係がある。

【0049】

ここで用いた、ある整数 y を法とする剰余演算子 $\bmod y$ は、非負整数 $x \geq 0$ に対しては C 言語などの剰余演算子 ' $\%$ ' と同じで

$$x \bmod y = x \% y$$

である。なお $\%$ 演算子は、負の整数 $x < 0$ に対しては、

$$x \% y = -((-x) \% y)$$

のように奇関数となるが、ここで用いる $\bmod y$ は、結果が負にならないよう

$$x \bmod y = (x \% y) + y \quad (x < 0 \text{ の場合})$$

であるとする。例えば

$$(-1) \bmod 10 = 9$$

である。

【0050】

さらに、

$$\begin{aligned} \uparrow t_A + \uparrow t_C + \uparrow t_I + \uparrow t_K &= [4, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 4] \\ &= 4 [1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1] \end{aligned}$$

となる。したがって、任意の整数要素ベクトル $\uparrow x$ に対し、 $A + C + I + K$ は常に 4 の倍数になる。

【0051】

つまり A, C, I の値がわかっている場合, K の 4 を法とする剰余と (A + C + I) のそれとの和は 0 か 4 となるので, K の存在範囲がわかり, そこに間隔 4 で存在することになる。

【0 0 5 2】

さらに以下のような関係もある。

【0 0 5 3】

$$\begin{aligned} \uparrow t_B - \uparrow t_J + \uparrow t_E - \uparrow t_G &= [0, 4, 4, 0, 4, 4, 0, -4, 4, 0, -4, -4, 0, -4, -4, 0] \\ &= 4 [0, 1, 1, 0, 1, 1, 0, -1, 1, 0, -1, -1, 0, -1, -1, 0] \end{aligned}$$

$$\begin{aligned} 2. \quad 5 (\uparrow t_A + \uparrow t_C + \uparrow t_I + \uparrow t_K) + 2 (\uparrow t_E + \uparrow t_G) + \uparrow t_M + \uparrow t_O \\ = [20, 0, 0, 20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \end{aligned}$$

$$= 20 [1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\begin{aligned} 2. \quad 5 (\uparrow t_A + \uparrow t_C + \uparrow t_I + \uparrow t_K) + 2 (\uparrow t_B + \uparrow t_J) + \uparrow t_D + \uparrow t_L \\ = [20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 20, 0, 0, 0] \end{aligned}$$

$$= 20 [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]$$

$$\begin{aligned} 6. \quad 25 (\uparrow t_A + \uparrow t_C + \uparrow t_I + \uparrow t_K) \\ + 5 (\uparrow t_B + \uparrow t_E + \uparrow t_G + \uparrow t_J) \\ + 2. \quad 5 (\uparrow t_D + \uparrow t_L + \uparrow t_M + \uparrow t_O) \\ + 4 \uparrow t_F + 2 (\uparrow t_H + \uparrow t_N) + \uparrow t_P \\ = [100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \\ = 100 [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \end{aligned}$$

つまり,

- 1) A の範囲を 1 間隔にループ
 - 2) C の範囲を 2 間隔にループ (A を利用)
 - 3) B の範囲を 2 間隔にループ (A, C を利用)
 - 4) D の範囲を 10 間隔にループ (A, B, C を利用)
 - 5) I の範囲を 2 間隔にループ (A を利用)
 - 6) E の範囲を 2 間隔にループ (A, I を利用)
 - 7) M の範囲を 10 間隔にループ (A, I, E を利用)
 - 8) K の範囲を 4 間隔にループ (A, C, I を利用)
 - 9) G の範囲を 2 間隔にループ (E を利用)
 - 10) F の範囲を 2 間隔にループ (E, G を利用)
 - 11) H の範囲を 10 間隔にループ (E, F, G を利用)
 - 12) J の範囲を 4 間隔にループ (B, E, G を利用)
 - 13) L の範囲を 20 間隔にループ (A, B, C, D, I, J, K を利用)
 - 14) N の範囲を 10 間隔にループ (B, F, J を利用)
 - 15) O の範囲を 20 間隔にループ (A, C, E, G, I, K, M を利用)
 - 16) P の範囲を 100 間隔にループ (A ~ O を利用)
- のように多重ループを実行すれば, 各 for ループを 1 間隔にループするのに比べ, 総ループ回数を

$$\begin{aligned} 1 / (2 * 2 * 10 * 2 * 2 * 10 * 4 * 2 * 2 * 10 * 4 * 20 * 10 * 20 * 100) \\ = 1 / 409,600,000,000 \end{aligned}$$

に削減できる。

【0 0 5 4】

具体的な手順はアルゴリズム 2 のようになる。ここで C'_{\min} , B'_{\min} , D'_{\min} は,

$$f(x, y) = (-x) \bmod y \quad (17)$$

を用いて例えば次のようにして求める。

【0 0 5 5】

$$C'_{\min} = C_{\min} + f(C_{\min} + A, 2) \quad (18)$$

$$B'_{\min} = B_{\min} + f(B_{\min} + (A >> 1) + (C >> 1), 2) \quad (19)$$

$D'_{\min} = D_{\min} + f(D_{\min} + 2B + 2.5(A + C), 10)$
 $E'_{\min} \sim P'_{\min}$ も同様である。

【0056】

(アルゴリズム2)

```

1.  index ← 0
2.  cases ← 0
3.  for A ← Amin to Amax (*ここだけは1間隔*)
4.  for C ← Cmin to Cmax step 2
5.  for B ← Bmin to Bmax step 2
6.  for D ← Dmin to Dmax step 10
7.  for I ← Imin to Imax step 2
8.  for E ← Emin to Emax step 2
9.  for M ← Mmin to Mmax step 10
10. for K ← Kmin to Kmax step 4
11. for G ← Gmin to Gmax step 2
12. for F ← Fmin to Fmax step 2
13. for H ← Hmin to Hmax step 10
14. for J ← Jmin to Jmax step 4
15. for L ← Lmin to Lmax step 20
16. for N ← Nmin to Nmax step 10
17. for O ← Omin to Omax step 20
18. for P ← Pmin to Pmax step 100
19.  begin
20.    R ← T-1X(Tt)-1 (*式(8)*)
21.    if Rの全要素が整数である
22.      then (*A~Pが整数信号の直交変換の結果として妥当*)
23.        U ← R + Y (*式(9)*)
24.        if Uの全要素が8bit整数の範囲【0~255】におさまっている
25.          then (*A~Pが残差信号の直交変換の結果として妥当*)
26.            if Xが原変換係数に一致する
27.              then index ← cases
28.            endif
29.          cases ← cases + 1
30.        endif
31.      endif
32.    end
33.  cases分の情報量を用いてindexを符号化

```

【第3の発明】：ループ多重度削減

上記第1の発明および第2の発明におけるアルゴリズムは、いずれも16重ループを用いるものであったが、第3の発明では、さらに高速化するためにループの多重度を減らすようにする。ここではA~Pの16係数のうち

A	B	C	D
E	.	.	.
I	.	K	.
M	.	.	.

の8係数を別途伝送することでこれらのループを除くようにする。

【0057】

まず係数Aの伝送を考える。例えば「付加情報」として

$$Z_A = A - A_{\min}$$

を伝送すれば、復号側で

$$A = Z_A + A_{\min}$$

とすることでAを復元することができる。

【0058】

なお、このときAのとり得る場合の数が

$$A_{\max} - A_{\min} + 1$$

であるので、 Z_A を符号化するのに必要な情報量は、

$$\log_2 (A_{\max} - A_{\min} + 1) \text{ [bit]} \quad (21)$$

となる。これは復号側でも共有できるので、 Z_A を復号することは可能である。

【0059】

B～PについてもこのAのように伝送することも可能であるが、残差信号は直交変換後の空間内で非常に疎に分布していることから、符号量が大変に無駄になる。そこで例えばBについては、A、Cが既知の場合Bに関するループを2間隔にできたことから、

$$Z_B = (B - B'_{\min}) / 2$$

を伝送すれば、復号側で

$$B = 2Z_B + B'_{\min}$$

によりBを復元できる。 Z_B を符号化するのに必要な情報量は、

$$\log_2 ((B_{\max} - B'_{\min}) / 2 + 1) \text{ [bit]}$$

となる。C～Kについても同様である。したがって、

1) Aの値を Z_A により符号化

2) Cの範囲を約1/2に狭めた Z_C で符号化(Aを利用)

3) Bの範囲を約1/2に狭めた Z_B で符号化(A、Cを利用)

4) Dの範囲を約1/10に狭めた Z_D で符号化(A、B、Cを利用)

5) Iの範囲を約1/2に狭めた Z_I で符号化(Aを利用)

6) Eの範囲を約1/2に狭めた Z_E で符号化(A、Iを利用)

7) Mの範囲を約1/10に狭めた Z_M で符号化(A、E、Iを利用)

8) Kの範囲を約1/4に狭めた Z_K で符号化(A、C、Iを利用)

のような手順を実行することで、Xのうち

A B C D

E . . .

I . K .

M . . .

の8係数が無駄なく伝送される。このように範囲を狭められた Z_X が、第3の発明の「該係数がとりえない値を除いた情報」に相当する。

【0060】

残る8係数

. . . .

. F G H

. J . L

. N O P

については、第1の発明と同様、まとめて1つの数indexで表現し伝送する。

【0061】

これらをまとめるとアルゴリズム3に示すような手順となる。

【0062】

{アルゴリズム3}

1. A, C, B, D, I, E, M, Kをこの順で符号化する

2. index ← 0

3. cases ← 0

4. for $G \leftarrow G'_{\min}$ to G_{\max} step 2

5. for $F \leftarrow F'_{\min}$ to F_{\max} step 2

6. for $H \leftarrow H'_{\min}$ to H_{\max} step 10

```

7.  for J ← J' min to J_max step 4
8.  for L ← L' min to L_max step 2 0
9.  for N ← N' min to N_max step 1 0
10. for O ← O' min to O_max step 2 0
11. for P ← P' min to P_max step 1 0 0
12. begin
13.  R ← T-1X (Tt)-1 (*式(8)*)
14.  if Rの全要素が整数である
15.    then (*A~Pが整数信号の直交変換の結果として妥当*)
16.      U ← R + Y (*式(9)*)
17.      if Uの全要素が8bit整数の範囲【0~255】におさまっている
18.        then (*A~Pが残差信号の直交変換の結果として妥当*)
19.          if Xが原変換係数に一致する
20.            then index ← cases
21.          endif
22.          cases ← cases + 1
23.        endif
24.      endif
25.    end
26.  cases分の情報量を用いてindexを符号化

```

上記のようにして、16重だったループが8重に削減され、処理がより高速になる。

【0063】

なおindexを記述するのに必要な情報量は

$\log_2 \text{cases} [\text{bit}]$ (22)

である。これは復号側が付加情報なしに持つことができ、かつindexの復号に必要な情報である。

【0064】

【第7の発明】：対応する復号

ここでは、上記第3の発明に係る符号化に対応する復号に係る発明について説明する。復号側でも符号化と同様のループを回すことにより、ループ最内部の妥当性判断の結果がindex+1回目に真となったときの、A~KおよびG~Pの値が、原残差信号に対応する直交変換係数となることを知る。

【0065】

indexの符号化時にこれがどれだけの情報量を持つかに応じて符号化した場合、indexの復号においては、まずこれがどれだけの情報量をもつかを知らなければならない。そこで予めcasesを得る必要がある。そのためにはやはり符号化と同様のループを実行することが必要である。

【0066】

例えば、メモリは余分に必要であるが、多重ループの二度実行を避けるためにアルゴリズム4のようにすることで多重ループを一度実行することによりcases, index, 原信号をこの順で取得できる。ここでUbuf[]は、4×4行列を要素とする配列であり、動的に確保あるいはあらかじめ十分な量を確保しておくこととする。

【0067】

〔アルゴリズム4〕

```

1. A, C, B, D, I, E, M, Kをこの順で復号する
2.  cases ← 0
3.  for G ← G' min to G_max step 2
4.  for F ← F' min to F_max step 2
5.  for H ← H' min to H_max step 1 0
6.  for J ← J' min to J_max step 4

```

```

7.  for L ← L' min to L_max step 2 0
8.  for N ← N' min to N_max step 1 0
9.  for O ← O' min to O_max step 2 0
10. for P ← P' min to P_max step 1 0 0
11.  begin
12.  R ← T-1X (Tt)-1 (*式(8)*)
13.  if Rの全要素が整数である
14.    then (*A～Pが整数信号の直交変換の結果として妥当*)
15.      U ← R + Y (*式(9)*)
16.      if Uの全要素が8 b i t 整数の範囲【0～255】におさまっている
17.        then (*A～Pが残差信号の直交変換の結果として妥当*)
18.          U b u f [cases] ← U
19.          cases ← cases + 1
20.        endif
21.      endif
22.    end
23.  cases分の情報量を用いてindex を復号
24. U ← U b u f [index] (*原信号値*)

```

【第6の発明】：高速な符号量推定

H. 264方式では、フレーム内予測においては複数のモードや種類、フレーム間予測においては複数のブロックサイズなど、さまざまな予測モードを選択することができる。予測残差信号Rも予測モードにより変化するため、符号量対歪みの意味で最適な予測モードを選択する場合、モード毎の符号量を適切に見積もる必要が生じる。

【0068】

可逆符号化の場合、歪みは常に零であるので、符号量（H. 264準拠の符号量と、付加情報の符号量の和）を最小化することだけに注意すればよい。

【0069】

アルゴリズム1，アルゴリズム2，アルゴリズム3に示したような多重ループを実行すれば当然正確な付加情報符号量を見積もることが可能であるが、より高速な符号量推定法があれば、複数のモードを比較するのに都合がよい。

【0070】

第6の発明は、多重ループを実行することなく、付加情報の符号量を見積もることを目的としている。

【0071】

まず第1の発明，第2の発明のように特定の変換係数を事前に伝送しない場合を説明する。事前に量子化前の係数がとり得る上下限值（式(13)，式(14)）を求めておく。これを元に第6の発明の「変換係数の存在空間」の体積Vを次のように求める：

【0072】

【数5】

$$V = \prod_{x \in A, \dots, P} (x_{\max} - x_{\min} + 1) \quad (23)$$

このVが存在空間中の格子点の数を近似していると考えられる。この中のすべてが原残差信号に対応しているわけではなく、空間は式(6)に示したように拡大されているため、Vをこの倍率で除したV₀

$$V_0 = V / 40^{4+4} \quad (24)$$

が格子点の総数を近似していると考えられる。したがって、1画素あたりの付加情報符号量l₀は、

$$\begin{aligned}
l_0 &= \log_2 V_0 / 16 \\
&= \log_2 (V / 40^{4+4}) / 16 \\
&= \log_2 V / 16 - 2.66 \text{ [bit]}
\end{aligned}
\tag{25}$$

となる。

【0073】

また第3の発明のように特定の変換係数を事前に伝送する場合、それらの情報量 l_1 は式(21)と同様にして、

【0074】

【数6】

$$l_1 = \sum_{x \in A, \dots, K} ((x_{\max} - x'_{\min}) / n_x + 1) \tag{26}$$

ここで x は

A	B	C	D
E	.	.	.
I	.	K	.
M	.	.	.

の係数である。

【0075】

なお n_x はすでに述べた区間を狭める量であり、A, ..., Mそれぞれについて

1	2	2	10
2	.	.	.
2	.	4	.
10	.	.	.

である。 $A'_{\min} = A_{\min}$ 、 $C'_{\min} \sim M'_{\min}$ は、式(18)などのようにして求める。

【0076】

また、残る8次元の「変換係数の存在空間」の体積 V_1 は次のようになる：

【0077】

【数7】

$$V_1 = \prod_{x \in F, \dots, P} (x_{\max} - x_{\min} + 1) \tag{27}$$

ここでも、先のケース同様、原残差信号に対応する格子点はまばらに存在している。この空間の拡大率は、式(6)を n_x ($x \in A, \dots, M$) をすべて掛け合わせたもので除した

【0078】

【数8】

$$\begin{aligned}
n_0 &= 40^{4+4} / \prod_{x \in A, \dots, M} n_x = 40^8 / (1 * 2 * 2 * 10 * 2 * 2 * 4 * 10) \\
&= 1,024,000,000
\end{aligned}
\tag{28}$$

となっているはずである。そこで「格子点の総数」の情報量 l_1 は一画素あたり

$$\begin{aligned}
l_2 &= \log_2 (V_1 / n_0) / 16 \\
&= \log_2 V_1 / 16 - 1.87 \text{ [bit]}
\end{aligned}
\tag{29}$$

となる。

【 0 0 7 9 】

結局，一画素あたり付加情報量は両者の和

$$l_1 + l_2 \text{ [b i t]}$$

(3 0)

となる。

【 0 0 8 0 】

式 (2 5)，式 (3 0) のように見積もられた値を利用して，多重ループを用いる符号化を実行することなしに，符号量の推定を行うことができる。これを用いて符号量最小となるモードを選択し，選択されたモードについてだけ実際の符号化を行うことで，準最適な可逆符号化を実現することができる。

【 0 0 8 1 】

〔第 4 の発明〕：妥当性判断の高速化

【 0 0 8 2 】

【数 9】

$$T^{-1} = \begin{bmatrix} 0.25 & 0.2 & 0.25 & 0.1 \\ 0.25 & 0.1 & -0.25 & -0.2 \\ 0.25 & -0.1 & -0.25 & 0.2 \\ 0.25 & -0.2 & 0.25 & -0.1 \end{bmatrix} \quad (31)$$

であるので，式 (8) をそのまま計算する場合，浮動小数点演算が必要になる。しかしながら，これを 2 0 倍すると，

【 0 0 8 3 】

【数 1 0】

$$T_0 = 20T^{-1} = \begin{bmatrix} 5 & 4 & 5 & 2 \\ 5 & 2 & -5 & -4 \\ 5 & -2 & -5 & 4 \\ 5 & -4 & 5 & -2 \end{bmatrix} \quad (32)$$

となるので， T^{-1} の代わりに T_0 を用いることで，以下のように正しい R の 4 0 0 (= 2 0 * 2 0) 倍の値 (R_{400} とする) を整数演算のみで得ることができる。

【 0 0 8 4 】

$$R_{400} = 400 R = T_0 \times T_0^t$$

(3 3)

ここで

【 0 0 8 5 】

【数 1 1】

$$\begin{bmatrix} s \\ t \\ u \\ v \end{bmatrix} = T_0 \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \quad (34)$$

としたとき，式 (3 3) は式 (3 4) を 8 回縦横に施すことに分解できるが，この計算は具体的にアルゴリズム 5 のように高速に求められる。

【 0 0 8 6 】

〔アルゴリズム 5〕

1. $x_0 \leftarrow 5(a + c)$
2. $x_1 \leftarrow 5(a - c)$
3. $x_2 \leftarrow 4b + 2c$
4. $x_3 \leftarrow 2b - 4c$
5. $s \leftarrow x_0 + x_2$
6. $t \leftarrow x_1 + x_3$
7. $u \leftarrow x_1 - x_3$
8. $v \leftarrow x_0 - x_2$

ここで、 $5x$ は $(x \ll 2) + x$ のようにシフト1回、加算1回で求められるし、 $4b + 2c$ なども $(b \ll 2) + (c \ll 1)$ としてシフト2回、加算1回で求められるため、式(33)の計算に乗算は一切不要である。

【0087】

ここで、前述した「妥当性判断」の一つである、「Rの全要素が整数であること」は、「 R_{400} の全要素が400の倍数であること」と等価である。これを実行する場合、400で除した余りが0であることを確認するために、除算が必要となる。しかしながら、既にループを回す時点で25の倍数である冗長性は除去されているので、16の倍数であることを確認すればよい。これは「 R_{400} の全要素の下位4bitがすべて0であること」と等価である。これは「各要素と15(二進数で1111)とのビットAND演算の結果が0であること」と等価であり、除算を用いずに判断ができる。

【0088】

ここで、もう一つの「妥当性判断」である、「Uの全要素が8bit整数の範囲[0～255]におさまっていること」も修正が必要になる。式(9)の両辺に400を乗ずると

$$U_{400} = 400U = R_{400} + Y_{400} \quad (35)$$

となる。ここで Y_{400} は予測信号Yを400倍したものであり、これを予め用意しておき(256要素のルックアップテーブルを用いれば400倍の乗算は不要である)、 R_{400} との和を求めれば、原信号の400倍が得られる。ここで $255 * 400 = 102,000$ なので「 U_{400} の全要素が[0～102000]におさまっていること」を確認すればよい。

【0089】

この手順をまとめるとアルゴリズム6のようになる。

【0090】

〔アルゴリズム 6〕

1. $index \leftarrow 0$
2. $cases \leftarrow 0$
3. for $A \leftarrow A_{min}$ to A_{max} (*ここだけは1間隔*)
4. for $C \leftarrow C'_{min}$ to C_{max} step 2
5. for $B \leftarrow B'_{min}$ to B_{max} step 2
6. for $D \leftarrow D'_{min}$ to D_{max} step 10
7. for $I \leftarrow I'_{min}$ to I_{max} step 2
8. for $E \leftarrow E'_{min}$ to E_{max} step 2
9. for $M \leftarrow M'_{min}$ to M_{max} step 10
10. for $K \leftarrow K'_{min}$ to K_{max} step 4
11. for $G \leftarrow G'_{min}$ to G_{max} step 2
12. for $F \leftarrow F'_{min}$ to F_{max} step 2
13. for $H \leftarrow H'_{min}$ to H_{max} step 10
14. for $J \leftarrow J'_{min}$ to J_{max} step 4
15. for $L \leftarrow L'_{min}$ to L_{max} step 20
16. for $N \leftarrow N'_{min}$ to N_{max} step 10

```

17. for O ← O' min to O_max step 2 0
18. for P ← P' min to P_max step 1 0 0
19. begin
20. R400 ← T0 X T0t (*式(33)*)
21. if R400 の全要素の下位 4 b i t が 0 である
22. then (*A ~ P が整数信号の直交変換の結果として妥当*)
23. U400 ← R400 + Y400 (*式(35)*)
24. if U400 の全要素が [0 ~ 1 0 2 0 0 0] におさまっている
25. then (*A ~ P が残差信号の直交変換の結果として妥当*)
26. if X が原変換係数に一致する
27. then index ← cases
28. endif
29. cases ← cases + 1
30. endif
31. endif
32. end
33. cases 分の情報量を用いて index を符号化

```

【第5の発明】：凸性を利用した処理高速化

残差信号

【0 0 9 1】

【数 1 2】

$$R = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix}$$

の存在空間を考える。左上の要素 a は、この位置の予測画素信号を Y_a とすると、

$$0 \leq a + Y_a \leq 255 \quad (36)$$

$$-Y_a \leq a \leq 255 - Y_a \quad (37)$$

という存在範囲を満たしている。b ~ p も同様に上限・下限が定まっている。16次元空間で考えると、Rの存在範囲Ωは凸多面体（超直方体）の形状をしていることになる。

【0 0 9 2】

Rの存在範囲を式(3)により変換した先の、やはり16次元の空間でも、回転・拡大を経ているが凸多面体であることには変わりがない。変換係数Xはこの変換先の空間内に存在しているが、各要素の上下限は式(13)、式(14)のように求まっているため、Xの存在範囲Ψもやはり凸多面体（超直方体）となっている。したがって、ΩとΨの交わりΩ∩Ψも、やはり凸多面体になる。これらの関係を簡単のため2次元で図示すると図1のようになる。

【0 0 9 3】

図1のΨ領域を拡大したものが図2である。格子点を列挙するステップにおいて多重ループを回す際、この図に示すように、A=A₀でBを変化させていった場合、妥当性判断の第二番（Uの全要素が8bit整数の範囲[0 ~ 255]におさまっている）は5個目の格子点（灰色）において「真」→「偽」となる。存在範囲Ω∩Ψが凸であるため、6個目以降は偽であることが確定するので、Bのループを中止し、次の、A=A₁としてBのループを再開する。ここでは6個目の格子点（灰色）において「真」→「偽」となるためループを中止する。このようにして、図の破線で囲った部分についての判断を中止することができる。

【0094】

これらをまとめるとアルゴリズム7のようになる。全係数について妥当性判断中止を行う手順を示すと字下げが深くなりすぎるため、ここでは簡単のため多重ループ最内部の3変数(N, O, P)について妥当性判断を中止する手順を示しているが、一般性を示すには十分である。

【0095】

{ アルゴリズム7 }

```

1.  index ← 0
2.  cases ← 0
3.  for A ← Amin to Amax
4.  for B ← Bmin to Bmax
5.  for C ← Cmin to Cmax
6.  for D ← Dmin to Dmax
7.  for E ← Emin to Emax
8.  for F ← Fmin to Fmax
9.  for G ← Gmin to Gmax
10. for H ← Hmin to Hmax
11. for I ← Imin to Imax
12. for J ← Jmin to Jmax
13. for K ← Kmin to Kmax
14. for L ← Lmin to Lmax
15. for M ← Mmin to Mmax
16.  begin
17.  CheckN ← 偽
18.  for N ← Nmin to Nmax
19.  begin
20.  CheckO ← 偽
21.  for O ← Omin to Omax
22.  begin
23.  CheckP ← 偽
24.  for P ← Pmin to Pmax
25.  begin
26.   $R \leftarrow T^{-1} X (T^t)^{-1} (*式(8)*)$ 
27.  if Rの全要素が整数である
28.  then (*A~Pが整数信号の直交変換の結果として妥当*)
29.   $U \leftarrow R + Y (*式(9)*)$ 
30.  if Uの全要素が8bit整数の範囲【0~255】におさまって
いる
31.  then (*A~Pが残差信号の直交変換の結果として妥当*)
32.  if Xが原変換係数に一致する
33.  then index ← cases
34.  endif
35.  Check N ← 真
36.  Check O ← 真
37.  Check P ← 真
38.  cases ← cases + 1
39.  else if CheckP = 真
40.  then goto 21 (*真→偽に変わった*)
41.  endif
42.  endif

```



```

43.         end
44.         if CheckP = 偽 and CheckO = 真
45.             then goto 18 (* 真→偽に変わった *)
46.         endif
47.     end
48.     if CheckP = 偽 and CheckO = 偽 and CheckN = 真
49.         then goto 15 (* 真→偽に変わった *)
50.     endif
51. end
52. end
53. cases分の情報量を用いてindex を符号化

```

【符号化装置の構成例】

上述した本発明を実現する装置構成について以下に説明する。図3は、本発明の段階的可逆ビデオ符号化装置の構成例を示す図である。段階的可逆ビデオ符号化装置1は、H.264標準符号化方式に基づき各ブロック毎に原画像信号と、フレーム内符号化における空間予測またはフレーム間符号化における時間予測による予測画像信号との残差信号 R_{orig} を算出する部分（図3では図示省略）の他に、cases、indexを初期化する初期化部11、残差信号 R_{orig} を直交変換する直交変換部12、直交変換部12の出力を量子化する量子化部13、上下限值情報 A_{min} 、 A_{max} 、 \dots 、 P_{min} 、 P_{max} を求める存在空間決定部14、逐次符号化伝送する係数を符号化処理する係数逐次符号化部15、係数を一括して代表する数indexを求めて符号化する係数一括符号化部16、画像内のすべてのブロックを符号化したかを判定する終了判定部17を備える。

【0096】

係数一括符号化部16は、各係数の上下限内ですべての係数の組合せを列挙する多重ループ開始部161、既に符号化伝送した係数および現在ループ中の係数により定まる 4×4 行列 X を逆直交変換する逆直交変換部162、行列和 $U = R + Y$ を求める予測信号加算部163、indexとcasesを更新処理する内部変数更新部164、各係数の上下限内ですべての係数の組合せが調べられたかを判定する多重ループ終了判定部165、indexを符号化する通し番号符号化部166から構成される。

【0097】

図3に示す段階的可逆ビデオ符号化装置1は、以下のように動作する。まず初期化部11において「格子点の総数」cases、残差信号の直交変換係数に一致する格子点の「通し番号」indexをそれぞれ0に初期化する。

【0098】

次いで直交変換部12において、残差信号 R_{orig} に直交変換を施し X_{orig} を得る。これを量子化部13において、量子化パラメータ情報を用いて量子化する。この量子化情報から、存在空間決定部14において、上下限值情報 A_{min} 、 A_{max} 、 \dots 、 P_{min} 、 P_{max} を求める。

【0099】

次に、係数逐次符号化部15で、逐次符号化伝送する係数を符号化処理する。例えば上述した第1の発明の場合0個、第3の発明の場合A、C、B、D、I、E、M、Kの8個である。

【0100】

次いで、係数一括符号化部16において、残る係数を一括して代表する数indexを求めて符号化する。すなわち、係数一括符号化部16中の多重ループ開始部161は、残る係数（例えば第1の発明の場合A～Pの16個、第3の発明の場合G、F、H、J、L、N、O、Pの8個）についてのそれぞれの上下限でとり得る範囲すべてを列挙する。その際、第2の発明のように、係数間の整数的値関係を利用して、とり得ない係数の組合せをとばして列挙してもよい。

【0101】

逆直交変換部 1 6 2 では、既に符号化伝送した係数および現在ループ中の係数 A ～ P により定まる 4×4 行列 X を逆直交変換し、R とする。ここでもし R に整数でない値が存在すれば、多重ループ終了判定部 1 6 5 に進む。

【0 1 0 2】

次に予測信号加算部 1 6 3 において、行列和 $U = R + Y$ を求める。ここでもし U の要素に $[0 \sim 255]$ の範囲に収まっていない値が存在すれば、多重ループ終了判定部 1 6 5 に進む。

【0 1 0 3】

次に内部変数更新部 1 6 4 において、X と X_{orig} が一致する場合、index に cases を代入する処理を行う。また、cases に 1 を加算する。

【0 1 0 4】

多重ループ終了判定部 1 6 5 では、上下限内ですべての係数の組合せが調べられたかを判定し、まだであれば多重ループ開始部 1 6 1 の処理へ戻る。済みであれば、通し番号符号化部 1 6 6 で、cases 分の情報量を用いて index を符号化する。

【0 1 0 5】

次いで、終了判定部 1 7 において、画像内のすべてのブロックを符号化したかを判定し、済んでいなければ次のブロックに移って初期化部 1 1 による処理を再開する。済みであれば、符号化を終了する。

【0 1 0 6】

〔index の符号化に汎用可変長符号化を用いる場合〕

本発明においては、例えば、汎用可変長符号（参考文献：Y. Itoh, N-M Cheung: "Universal variable length code for DCT coding", IEEE Proc. Int. Conf. Image Processing, Vol. 1, pp. 940-943, 2000）等を用いることにより、cases を使わずに index を符号化してもよい。この場合、内部変数更新部 1 6 4 において X と X_{orig} が一致すれば、その時の cases の値を汎用可変長符号化し、すぐさま係数一括符号化部 1 6 による処理を終了する。

【0 1 0 7】

〔復号装置の構成例〕

図 4 は、本発明の段階的可逆ビデオ復号装置の構成例を示す図である。段階的可逆ビデオ復号装置 2 は、量子化パラメータ・量子化係数を復号するとともに予測信号 Y の生成を行う初期化復号部 2 1、上下限值情報 A_{\min} , A_{\max} , ..., P_{\min} , P_{\max} を求める存在空間決定部 2 2、逐次符号化伝送された係数を復号する係数逐次復号部 2 3、係数を一括して代表する数 index を復号する係数一括復号部 2 4、U の値を記憶する配列記憶である U b u f [] 2 5、および画像内のすべてのブロックを復号したかを判定する終了判定部 2 6 から構成される。

【0 1 0 8】

係数一括復号部 2 4 は、各係数の上下限内ですべての係数の組合せを列挙する多重ループ開始部 2 4 1、既に復号された係数および現在ループ中の係数により定まる 4×4 行列 X を逆直交変換する逆直交変換部 2 4 2、行列和 $U = R + Y$ を求める予測信号加算部 2 4 3、U b u f [] 2 5 の cases 番地に U を格納するとともに cases を更新する内部変数更新部 2 4 4、各係数の上下限内ですべての係数の組合せが調べられたかを判定する多重ループ終了判定部 2 4 5、index を復号する通し番号復号部 2 4 6、U b u f [] 2 5 の index 番目の値を原画像信号として出力する原信号出力部 2 4 7 から構成される。

【0 1 0 9】

図 4 に示す段階的可逆ビデオ復号装置 2 は、以下のように動作する。まず初期化復号部 2 1 において、格子点の総数 cases を 0 に初期化し、量子化パラメータ・量子化係数の復号、予測信号 Y の生成を行う。

【0 1 1 0】

量子化パラメータと量子化係数とに基づき、存在空間決定部 2 2 において、上下限值情報 A_{\min} , A_{\max} , ..., P_{\min} , P_{\max} を求める。

【0 1 1 1】

次に、係数逐次復号部 2 3 で、逐次符号化伝送された係数を復号する。例えば上記第 1 の発明の場合 0 個、第 3 の発明の場合 A, C, B, D, I, E, M, K の 8 個をこの順に復号する。

【0 1 1 2】

次いで、係数一括復号部 2 4 において、残る係数を一括して代表する数 index を復号し可逆復元を行う。すなわち、係数一括復号部 2 4 中の多重ループ開始部 2 4 1 は、残る係数（例えば第 1 の発明の場合 A ~ P の 1 6 個、第 3 の発明の場合 G, F, H, J, L, N, O, P の 8 個）についてのそれぞれの上下限でとり得る範囲すべてを列挙する。

【0 1 1 3】

その際、第 2 の発明のように、係数間の整数的値関係を利用して、とり得ない係数の組合せをとばして列挙してもよい。ただし、とり得ない係数の組合せをとばして列挙する手順は、対応する符号化装置と全く同一の手順とする。

【0 1 1 4】

逆直交変換部 2 4 2 では、既に復号された係数および現在ループ中の係数 A ~ P により定まる 4×4 行列 X を逆直交変換し、R とする。ここでもし R に整数でない値が存在すれば、多重ループ終了判定部 2 4 5 の処理に進む。

【0 1 1 5】

次に、予測信号加算部 2 4 3 において、行列和 $U = R + Y$ を求める。ここでもし U の要素に $[0 \sim 255]$ の範囲に収まっていない値が存在すれば、多重ループ終了判定部 2 4 5 の処理に進む。

【0 1 1 6】

次に、内部変数更新部 2 4 4 において、予め確保しておいた U b u f [] 2 5 の cases 番地に U を格納し、cases に 1 を加算する。

【0 1 1 7】

多重ループ終了判定部 2 4 5 では、上下限内ですべての係数の組合せが調べられたかを判定し、まだであれば多重ループ開始部 2 4 1 による処理へ戻る。済みであれば、通し番号復号部 2 4 6 で、cases 分の情報量を用いて index を復号する。

【0 1 1 8】

次いで原信号出力部 2 4 7 で、U b u f [] 2 5 の index 番目の値を取り出し、原画像信号として出力する。

【0 1 1 9】

次いで、終了判定部 2 6 において、画像内のすべてのブロックを復号したかを判定し、済んでいなければ次のブロックに移って初期化復号部 2 1 による処理を再開する。済みであれば、復号を終了する。

【0 1 2 0】

〔index の符号化に汎用可変長符号化を用いる場合の復号装置の構成例〕

本発明において、index の符号化に汎用可変長符号を用いる場合、復号側は図 5 に示すような機能ブロック構成になる。図 5 に示す段階的可逆ビデオ復号装置 3 は、汎用可変長復号部 2 7 において index を復号し、多重ループ終了判定部 2 8 5 では多重ループを回った回数が index 回目であるかを判定し、そうであればすぐさま終了と判定し、そのときの U の値を原信号として出力する。したがって、図 4 に示す U b u f [] 2 5 のような配列記憶は不要となる。他の部分の動作は、図 4 で説明した例と同様である。

【実施例 1】

【0 1 2 1】

第 3 の発明に対応する実施例 1 について説明する。この実施例 1 において核となる処理の概略は以下となる。

【0 1 2 2】

入力：量子化パラメータ、予測信号 Y、原残差信号 R_{orig}

処理：原信号に対応する直交変換係数について、一部は圧縮情報化し、残りは、原信号

となり得る組合せをもれなくすべて列挙した中の通し番号として表現する。

【0123】

出力：係数の一部（A，C，B，D，I，E，M，K）をそれぞれ表現する圧縮情報（ $Z_A \sim Z_K$ ），残りの係数（G，F，H，J，L，N，O，P）をまとめて1数値で代表する符号（index）

実施例1について，図6に示す処理フローを参照して説明する。まずステップS101にて量子化パラメータ，予測信号Y，原残差信号 R_{orig} を入力する。次いでステップS102においてindex，casesをとともに0に初期化する。

【0124】

次いでステップS103において R_{orig} に直交変換を施し原変換係数 X_{orig} を得ると同時に， X_{orig} に量子化を施した量子化係数も求める。ステップS104では，これまでに得られた量子化パラメータ，量子化係数の情報から， X_{orig} を逆に推定した行列である，Xの各要素（A～P）の上下限を求める。これを元に，ステップS105において，A，C，B，D，I，E，M，Kをこの順で圧縮符号化する。

【0125】

続いて，G，F，H，J，L，N，O，Pの8重ループ（ステップS106～ステップS114）に入る。ステップS107にて，現在のA～Pの値からなる 4×4 行列Xを式（8）により逆変換しRを得る。

【0126】

ステップS108にて，Rの要素がすべて整数であることを確認し，偽であればステップS114へ，真であればステップS109に進み， $U = R + Y$ を計算する。

【0127】

次いでステップS110にて，Uの全要素が【0～255】の範囲内であるかどうかを確認する。これが偽であればステップS114へ，真であればステップS111に進み，Xが原変換係数 X_{orig} に等しいかを確認する。これが真であればステップS112にて，変数indexに現在のcasesの値を保存する。次いでステップS113にてcasesの値を1加算しステップS114へ進む。

【0128】

ステップS114では8重ループがすべて完了したかを確認し，偽である間はループ先頭のステップS106へ戻る。真であれば，ステップS115において，既に求められたcases（格子点の総数）分の情報量（式（22））を用いてindexを符号化し，終了する。

【0129】

ここでは，予め一部の係数（A，C，B，D，I，E，M，Kの8係数）をまず別途符号化する（ステップS105）処理例を示したが，これを省略し，ステップS106からステップS114までを全16係数の多重ループとすれば，全く同じ枠組で第1の発明または第2の発明に対応する実施例となる。

【実施例2】

【0130】

第7の発明に対応する実施例2について説明する。この実施例2において核となる処理の概略は以下となる。

【0131】

入力：量子化パラメータ，予測信号Y，量子化係数

処理：原信号に対応する直交変換係数について，一部（A，C，B，D，I，E，M，K）は圧縮情報から，残り（G，F，H，J，L，N，O，P）は，原信号となりうる組合せをもれなくすべて列挙した中の通し番号番目として復号する

出力：原画像信号

実施例2について，図7に示す処理フローを参照して説明する。まずステップS201にてcasesを0に初期化する。次いでステップS202において量子化パラメータ・量子化係数の復号，予測信号Yの生成を行う。これらはH. 264基本情報ビットストリーム

から得られる情報である。

【0132】

ステップS203では、これまでに得られた情報から、行列Xの各係数(A～P)の上下限を求める。これを元に、ステップS204において、A, C, B, D, I, E, M, Kをこの順で復号する。

【0133】

次いでG, F, H, J, L, N, O, Pの8重ループ(ステップS205～ステップS212)に入る。ステップS206にて、現在のA～Pの値からなる4×4行列Xを第8式により逆変換しRを得る。

【0134】

ステップS207にて、Rの要素がすべて整数であることを確認し、偽であればステップS212へ、真であればステップS208に進み、 $U=R+Y$ を計算する。

【0135】

次いでステップS209にて、Uの全要素が【0～255】の範囲内であるかどうかを確認する。これが真であれば、ステップS210にて、配列Ubuf[]のcases番地に現在のUの値を保存し、ステップS211へ進みcasesを1加算し、ステップS212へ進む。偽であればステップS212へ進む。

【0136】

ステップS212では8重ループがすべて完了したかを確認し、偽である間はループ先頭のステップS205へ戻る。真であれば、ステップS213において、既に求められたcases(格子点の総数)分の情報量(式(22))を用いてindexを復号し、ステップS214にてUbuf[index](原信号に相当)を出力し、終了する。

【0137】

以上の実施例では、casesを用いてindexを符号化する例を説明したが、indexの符号化に汎用可変長符号化(UVLC: Universal Variable Length Code)やストップ符号を用いるようにしてもよい。indexの符号化に汎用可変長符号化やストップ符号を用いた場合、復号側ではcasesがわからなくても復号することができる。この場合、index用の符号量は2割増えたと予想されるが、casesの値を算出する必要がないため、多重ループを途中で打ち切ることができ、符号化も復号も時間が約1/2になるというメリットがある。また、復号側の配列記憶Ubuf[]が不要になるというメリットもある。

【図面の簡単な説明】

【0138】

【図1】R, Xの存在範囲 Ω と Ψ をそれぞれ図示したものである。

【図2】 Ψ 領域の拡大図である。

【図3】段階的可逆ビデオ符号化装置の構成例を示す図である。

【図4】段階的可逆ビデオ復号装置の構成例を示す図である。

【図5】indexの符号化に汎用可変長符号を用いた場合の段階的可逆ビデオ復号装置の構成例を示す図である。

【図6】実施例1の処理フローを示す図である。

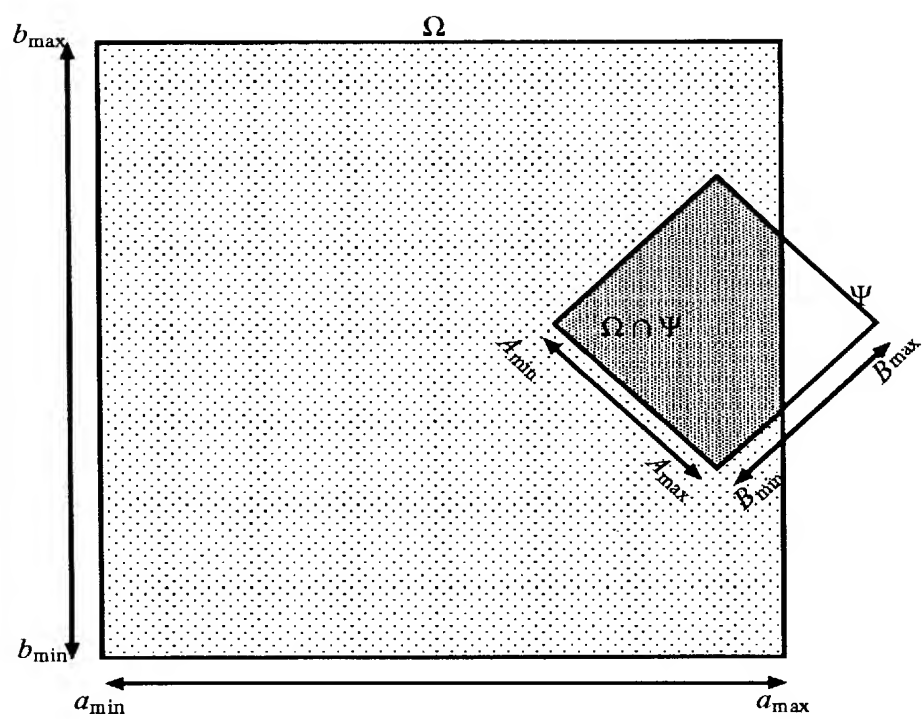
【図7】実施例2の処理フローを示す図である。

【符号の説明】

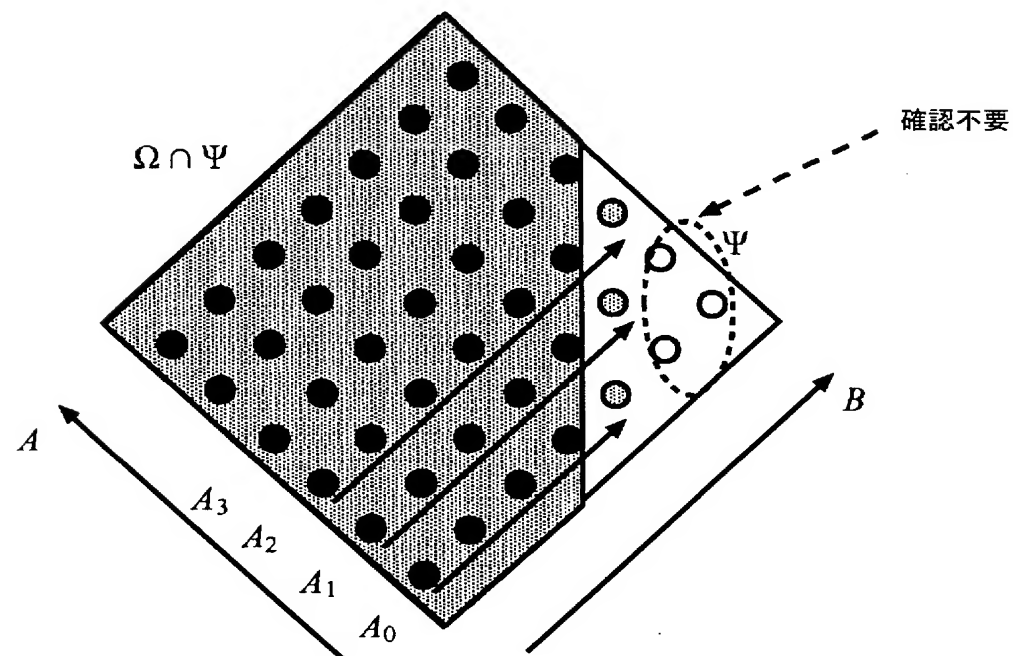
【0139】

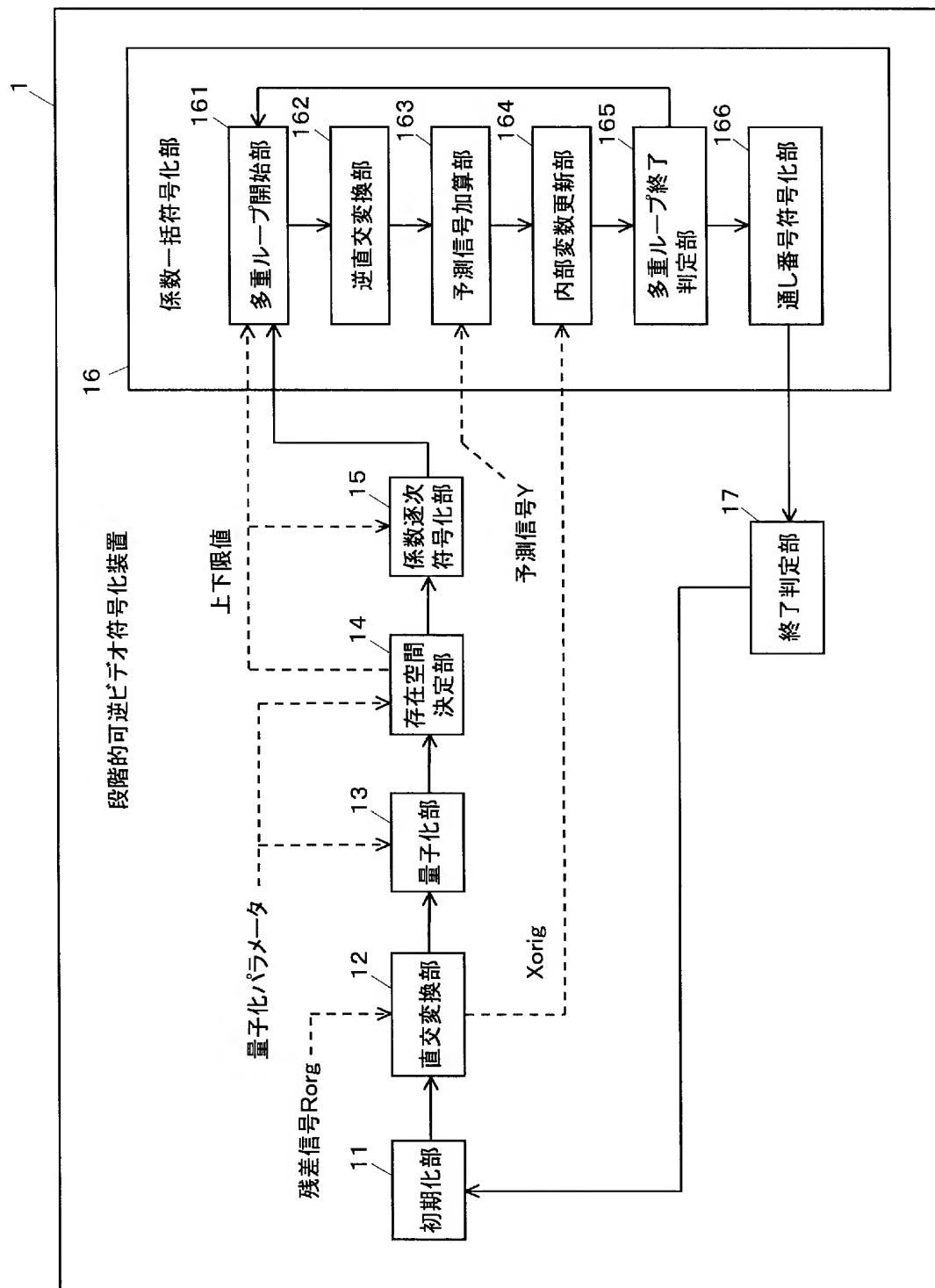
- 1 段階的可逆ビデオ符号化装置
- 2, 3 段階的可逆ビデオ復号装置
- 11 初期化部
- 12 直交変換部
- 13 量子化部
- 14 存在空間決定部
- 15 係数逐次符号化部
- 16 係数一括符号化部

1 7 終了判定部
2 1 初期化復号部
2 2 存在空間決定部
2 3 係数逐次復号部
2 4 , 2 8 係数一括復号部
2 5 U b u f []
2 6 終了判定部
2 7 汎用可変長復号部
1 6 1 , 2 4 1 , 2 8 1 多重ループ開始部
1 6 2 , 2 4 2 , 2 8 2 逆直交変換部
1 6 3 , 2 4 3 , 2 8 3 予測信号加算部
1 6 4 , 2 4 4 , 2 8 4 内部変数更新部
1 6 5 , 2 4 5 , 2 8 5 多重ループ終了判定部
1 6 6 通し番号符号化部
2 4 6 通し番号復号部
2 4 7 , 2 8 6 原信号出力部

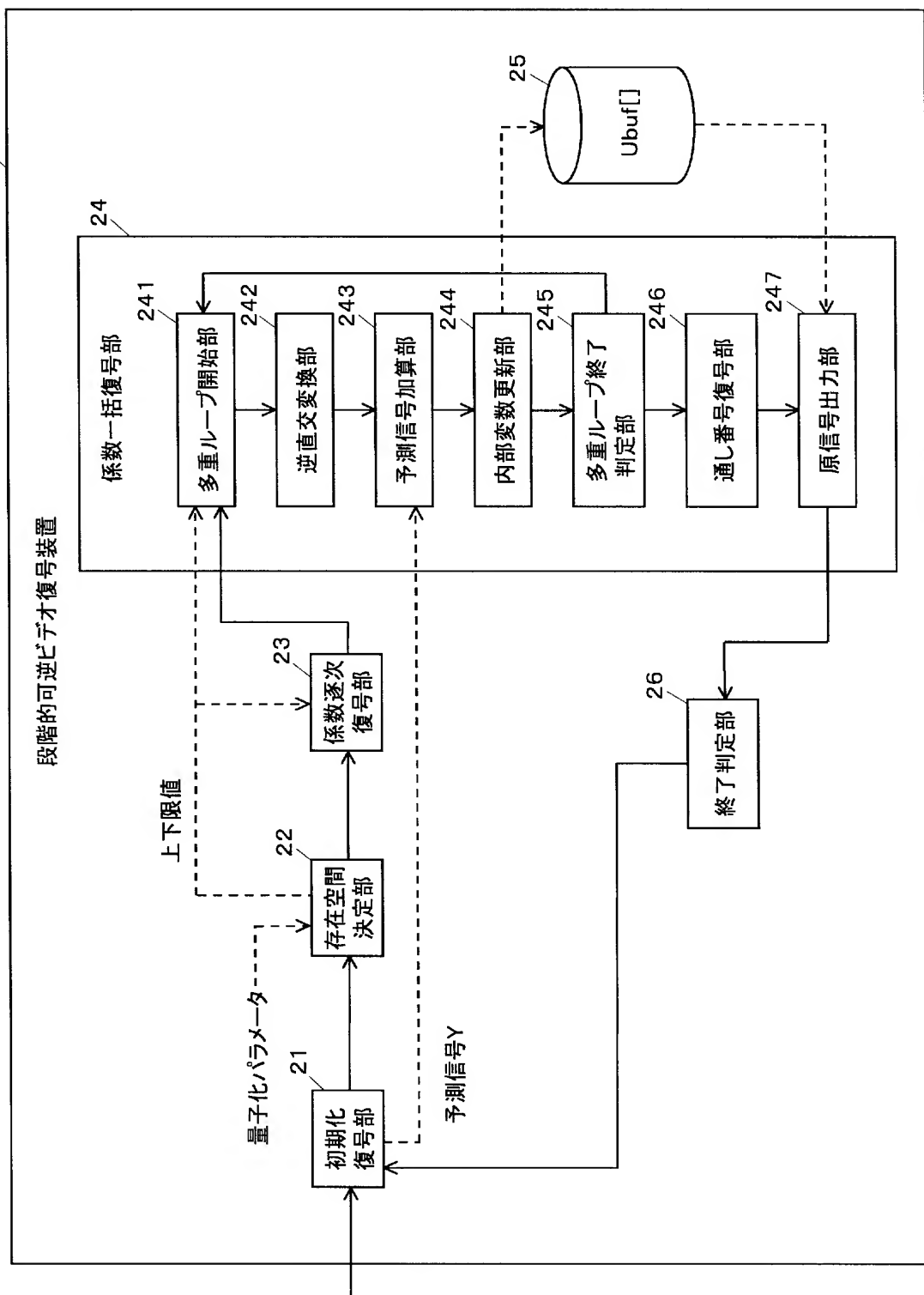


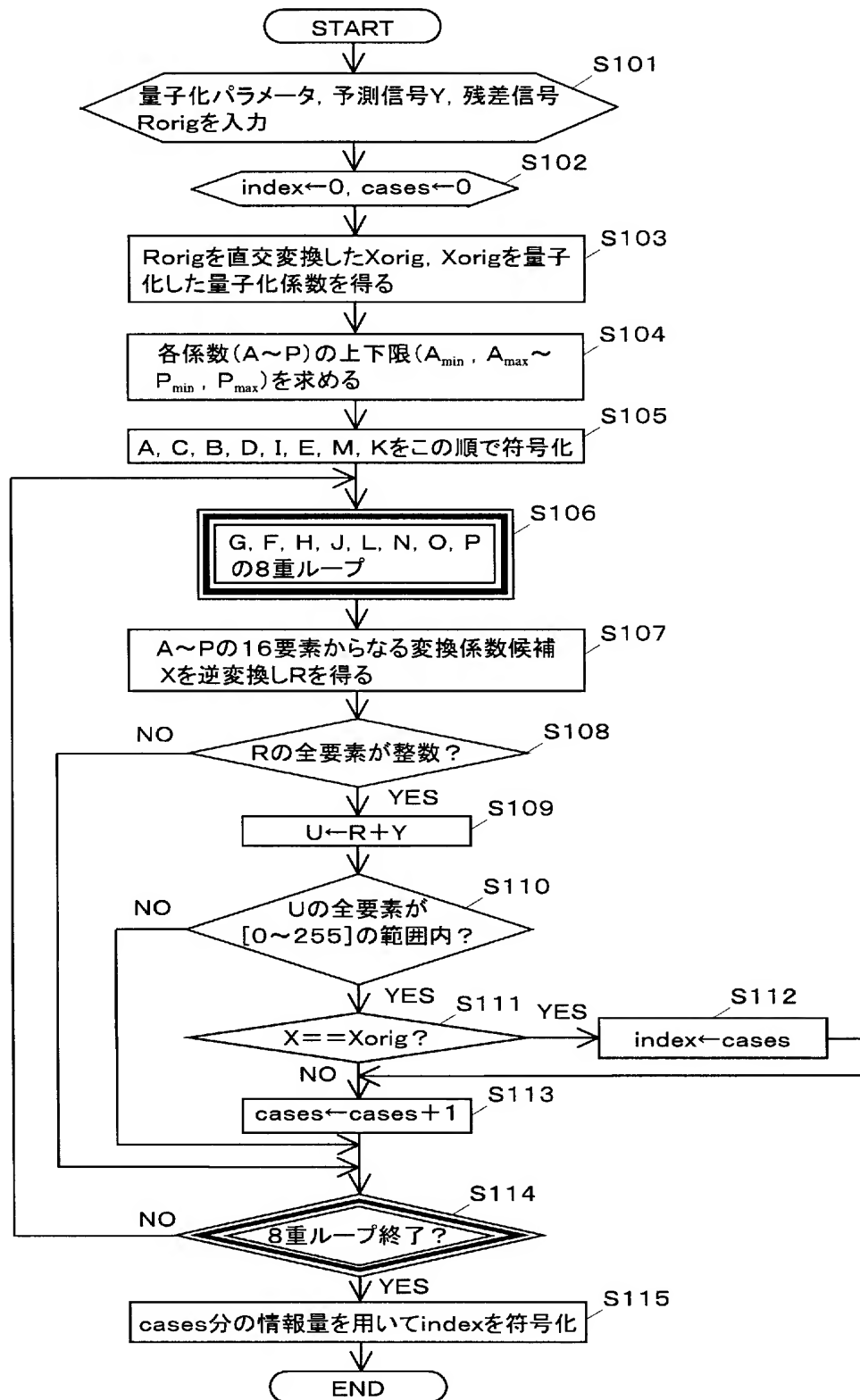
【図 2】

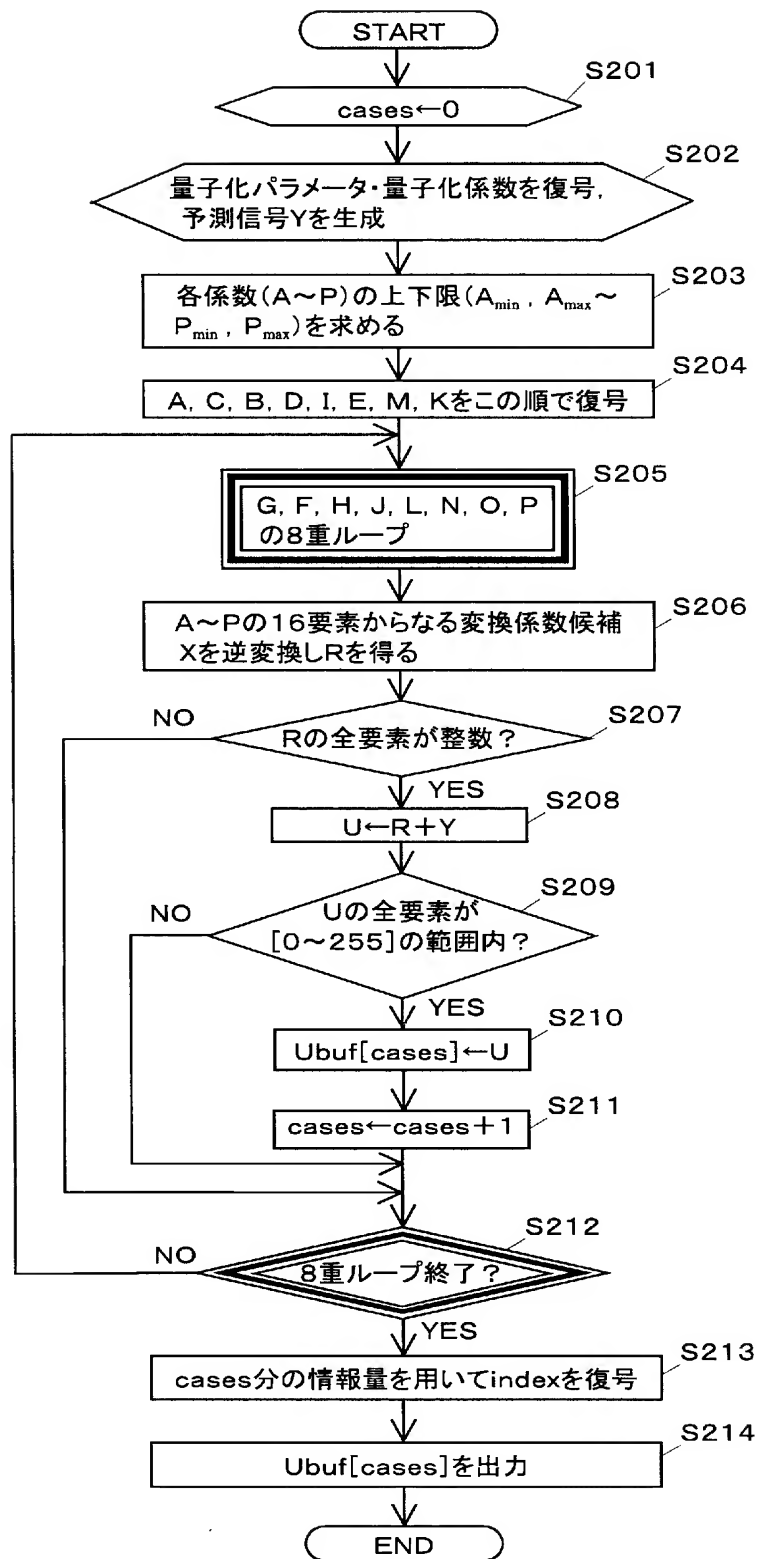




2







【書類名】要約書

【要約】

【課題】基本部分として伝送される符号はH. 264標準と互換性を保ちながら、高効率の可逆な復号を可能とする。

【解決手段】直交変換部12が残差信号 R_{orig} を直交変換して変換係数 X_{orig} を取得し、それを量子化部13が量子化する。存在空間決定部14は、量子化情報から各係数の上下限值情報（変換係数の存在空間）を求める。係数一括符号化部16において、変換係数の存在空間内の格子点が残差信号の直交変換の結果として妥当であるかどうかを判断し、妥当である格子点を列挙する。列挙順に通し番号(index)を割り当て、残差信号の変換係数 X_{orig} に一致する格子点の通し番号を通し番号符号化部166により符号化する。

【選択図】図3

出願人履歴

0 0 0 0 0 4 2 2 6

19990715

住所変更

5 9 1 0 2 9 2 8 6

東京都千代田区大手町二丁目3番1号

日本電信電話株式会社